An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 42 (2001) 1025–1033
www.elsevier.nl/locate/camwa

Numerical Experiences of Solving Elasticity Systems by PCG Methods

I. ARANY
Department of Applied Mathematics, University of Miskolc
H-3515 Miskolc, Hungary
matarany@gold.uni-miskolc.hu

Abstract—In solving systems of linear equations derived from FEM structural analysis by the conjugate gradient method, an efficient way for preparing effective incomplete factorization preconditioner was published by Saint-Georges *et al.* [1]. In [2,3], we prepared several variants of the cited method due to Saint-Georges *et al.* [1], and some of the relevant PCG-solvers were found more efficient than the original ones [1]. Here, we prepare some additional variants for producing good enough preconditioners by applying our spiral ordering [2] and execute some incomplete factorizations (MIC, RIC) with the level of fill-in dropping strategy which was found the best in [3]. In the comparison of the considered solvers, we analyze our computer results when systems from elastic frame structures with 3D beam elements were solved.

Keywords—Preconditioned conjugate gradient method, Incomplete factorization preconditioner, Spiral ordering, Level of fill-in dropping strategy.

1. INTRODUCTION

In solving systems of linear equations with a large sparse symmetric positive definite coefficient matrix, a widely used efficient way is to apply the preconditioned conjugate gradient (PCG) method with an effective preconditioner. In cases when the coefficient matrix is also an M-matrix, the preconditioner is usually prepared by an incomplete factorization [4,5]. That is, in a class of problems, for instance, when finite difference approximation is used for solving a Dirichlet problem for partial differential equations of elliptic type, the PCG method with an incomplete factorization preconditioner became an efficient and popularly used way. However, in case of systems of FEM structural analysis, a stiffness matrix is symmetric and positive definite, but usually, it is not an M-matrix, so, in trying to prepare an incomplete factorization preconditioner, usually it cannot be executed, since certain diagonal entry may get too close to zero during the process. For trying to avoid this problem, different approaches were published [6,7]; however, no general solution was found to this unpleasant problem. However, Saint-Georges *et al.* [1] published an excellent way for preparing a highly effective incomplete factorization preconditioner from a stiffness matrix by which some efficient PCG-solvers became available also for systems from FEM structural analysis.

The work was partly supported by funds AKP 97-146 2,3/31, FKFP 1040/1997, and OTKA T025172.
I wish to express my grateful thanks to T. Szabó (Department of Mechanics) for his useful advice and help during my preparation of this paper.

0898-1221/01/$ - see front matter Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(01)00218-8

As known, they execute a so-called DC-reduction by which a block-diagonal M-matrix is formed from the stiffness matrix. For improving the quality of the preconditioner to be prepared from it by an incomplete factorization [8], an ordering is executed before preparing an incomplete factorization preconditioner. In varying both the ordering and the incomplete factorization in the cited process, some additional PCG-solvers may be formed. In [2,9], we considered some variants of the above method for preparing the preconditioner and the most effective version was found, when, in the ordering phase, we applied our spiral ordering [10] which is available also for a stiffness matrix and we prepared the IC(0) preconditioner. In [3,11], we considered the variant for which we applied the above mentioned spiral ordering, and during the incomplete factorization, we applied the level of fill-in dropping strategy [12]. Testing this method with incomplete factorization of order IP (with parameter $0 \leq IP \leq 10$), in case of $IP = 2$, this version was found somewhat more efficient than that in [2,9].

Here we consider four further variants of the PCG-solvers. In the first two, after executing our spiral ordering, we prepare the incomplete factorization preconditioners, both as MIC(0) [5,13,14] and as RIC(0) [15,16]. In the third and fourth, we start to prepare the variant which was found the best in [11]. That is, we execute the spiral ordering, and during the incomplete factorization we apply the level of fill-in dropping strategy with $IP = 2$, while we prepare MIC and RIC preconditioners, respectively. (In other words, we apply MIC(2) and RIC(2).) For evaluating the considered PCG-solvers including also the ones in which IC(0) and IC(2) preconditioners are used, we solve several systems of linear equations from the structural analysis, when bar structure with 3D beam elements are solved. For measuring the efficiency of a PCG-version, we indicate the number of iterations required for achieving a given relative iterative error. For the two best versions, we consider how the number of iteration is changed while decreasing the relative iterative error.

2. INCOMPLETE FACTORIZATION PRECONDITIONERS

In solving the system of linear equations $Ax = b$ ($A \in R^{n \times n}$ is a sparse symmetric positive definite matrix, $b \in R^n$) by a PCG method, the number of iterations required for achieving the relative iterative error ε is $\lfloor 1/2\{\kappa(C^{-1}A)\}0.5 \ln(2/\varepsilon) + 1 \rfloor$, where $\kappa(.)$ is the spectral condition number of the relevant matrix, $C \in R^{n \times n}$ is the preconditioning matrix. As known, if $A \in R^{n \times n}$ is also an M-matrix, then a good enough C usually can be obtained by preparing an incomplete factorization of A (see [4,5]).

The simplest version of the incomplete factorization is when no fill-in is accepted during the factorization process (denoted by IC(0)). Another simple variant is to prescribe the number of fill-ins to be accepted in a row. In both cases, the data management in the implementation is very simple, since the storage requirement is predictable.

For well-structured matrices derived from a finite difference approximation, ones apply the strategy to define the positions of the fill-ins to be accepted in the factored matrix. For instance, let the accepted fill-ins be placed parallel to the main diagonal in the L factor and all the other fill-ins are discarded [13]. Ones use a drop tolerance principle [12], when discarding a fill-in element depends on its magnitude in absolute value. Then, the number of fill-ins becomes unpredictable by which the data management becomes more complicated. An improved variant of its conception was described by Saad [17].

For unstructured grid problems, ones apply some ordering for reducing the size of the envelope of the ordered matrix and a left-most nonzero entry in a row (defining the border of envelope) is discarded, if it is followed by a great number of zero entries [18]. The size of the truncated envelope obtained in such a way is reduced based on which a preconditioner is prepared.

A modified incomplete factorization (MIC) was introduced in [5,13], when all the discarded elements were added to the relevant diagonal entry. For preparing a more accurate approximate factorization, the first- and higher-order factorization were introduced by Gustafsson [13] for

well-structured matrices derived from the discretization of boundary value problems of partial differential equation, when finite difference approximation is applied. Then, fill-ins placed in the positions parallel to the main diagonal are accepted. As known, MIC(d) ($d \geq 0$) means that L contains d more nonzero subdiagonals than the lower part of A. For low-order factorizations, a nice reduction in condition number was achieved [5,13]. The conception of the above MIC(d) was generalized by Watts [12] by introducing the term of level of fill-in. Then, a parameter p ($p \geq 0$) defines the number of "subdiagonals" consisting of fill-ins in a general sparse symmetric matrix. That is, for each matrix element, we register its level of fill-in value as follows. At the beginning, as an initialization, for each nonzero entry, the level of fill-in value is zeroized. If in a position (i, j) a fill-in would occur, its numerical value is obtained as

$$F_{i,j} = -L_{i,k}U_{k,j},$$

where L and U denote the lower and upper parts of the factored matrix, respectively. The level of fill-in value for $F_{i,j}$ is defined as

$$\text{level}\,(F_{i,j}) = \text{level}\,(L_{i,k}) + \text{level}\,(U_{k,j}) + 1.$$

Obviously, if we have, for instance, level$(F_{i,j}) = 1$, then $F_{i,j}$ is placed in the first "subdiagonal" in the factored matrix. On the other hand, if for a fill-in element $F_{i,j}$ we have

$$\text{level}\,(F_{i,j}) > IP,$$

where IP is a prescribed parameter, then neglecting this fill-in entry and reset level$(F_{i,j}) = 0$, a dropping strategy is obtained. Here, the number of "subdiagonals" in the factored matrix is IP.

A relaxed incomplete factorization (RIC) was introduced by Axelsson *et al.* [15]. Then, if a fill-in element (to be discarded) would occur in the position (i, j), then its modified value (prepared by a relaxation parameter ω ($-1 \leq \omega < 1$)) is added to the diagonal entry both in row i and in row j. As it is discussed in [15,16], the rate of convergence of the RIC method is much better than that is for MIC method.

In [15,16], the dynamic versions of MIC and RIC (denoted as DMIC and DRIC) are discussed. For DMIC, in the pivot row, if the row sum in the upper triangular part divided by the diagonal entry is found to be larger than a given value τ with ($0 < \tau < 1$), then the diagonal entry will be changed and it will take this larger value. Here, $1 - \tau = h_0$, where $h_0 \approx 1/\sqrt[d]{\text{nuno}}$, where nuno denotes the number of nodes; d is the spatial dimension.

3. THE EFFECTS OF ORDERINGS ON THE CONVERGENCE OF A PCG METHOD

As known from the sparse matrix computation, in applying an ordering before a sparse symmetric factorization, the number of fill-ins may be remarkably changed. By choosing an appropriate ordering, both the storage- and the work-requirement of a direct method may become pleasantly small. Consequently, the effect of orderings on the quality of an incomplete factorization preconditioner used in a PCG method has great importance [8,19]. Duff *et al.* [8] considered a great number of orderings for demonstrating their effects on the convergence of the relevant PCG method, when incomplete factorization preconditioners were used. For each ordering, the tested domain was a 30×30 regular grid on which the Dirichlet problem for two-dimensional Laplacian equation by finite difference approximations was solved. The test results—among others—show that applying the spiral ordering, a nice convergence was achieved in most tested cases. As known, its starting point is a central point of the square regular mesh, and it can be determined easily for such a well-structured matrix. In [10], we presented an algorithm for finding a node of nearly minimum eccentricity of a general undirected graph. Note, its resulting node is usually exactly a central point of the graph. In such a way, applying this algorithm, a good starting point may be determined for general undirected graphs based on which a version of the spiral ordering became available also for the graphs of stiffness matrices.

4. THE GENERAL ALGORITHM FOR PREPARING EFFECTIVE PRECONDITIONERS

As mentioned, for systems of linear equations from the FEM structural analysis, it is often impossible to prepare any incomplete factorization preconditioner, since the stiffness matrix is sparse symmetric positive definite, but usually, it is not an M-matrix. Saint-Georges *et al.* in their excellent work [1] presented an effective way for preparing a highly efficient preconditioner for stiffness matrices and its essence is summarized as follows.

- Based on the mechanical properties of the whole problem, a so-called D-reduction is performed on the stiffness matrix by which a reduced matrix is obtained. That is, in each row, matrix entries of the same degree of freedom are grouped and all the others neglected. The reduced matrix obtained in such a way has a block-diagonal structure. The idea to produce such a reduced matrix was published first by Axelsson and Gustafsson [20] for 2D membrane problem. For 3D solid structure, Schlafman and Efrat [21] showed how to prepare also a block-diagonal matrix from the stiffness matrix. A similar approach is presented also in [22]. However, this reduced matrix usually is not an M-matrix.
- Applying the diagonal compensation [23] or C-reduction [24] on the obtained reduced matrix, an M-matrix is formed.

The above two steps together are called as DC-reduction. It is also shown in [1] that the spectral equivalence for the DC-reduction is assured.

- To improve the quality of the preconditioner to be prepared from the obtained M-matrix, an ordering is applied on it.
- An incomplete factorization is executed for preparing the preconditioner.

Note, a detailed analysis on this approach is discussed also in [25].

The skeleton of this original general algorithm in [1] for preparing an effective incomplete factorization preconditioner is as follows.

(1) Let us apply the DC-reduction to the stiffness matrix, by which a symmetric positive definite block-diagonal M-matrix is prepared from it.
(2) For improving the efficiency of the preconditioner to be prepared, an ordering is applied on the graph of the obtained M-matrix.
(3) For preparing the preconditioner, an incomplete factorization is executed.

Remarks

(i) In (2), the starting point is chosen as a node of maximum degree, and in the numbering generated we found the same (on our test problems) as that obtained by applying the Cuthill-McKee's numbering and reversing it.
(ii) In (3), each of the incomplete factorizations (IC, MIC, DMIC, RIC, and DRIC) was applied by which a number of PCG-solvers have became available.

5. THE CONSIDERED PCG-SOLVERS

In changing either the ordering or the incomplete factorization (or both) in the original algorithm in [1], we can get new preconditioners, by which new versions of PCG-solvers may be formed.

As mentioned, the most efficient version we presented in [9] was found better than those from [1]. On the other hand, the most efficient version of PCG-solver we presented in [11] was found to be better than any one in [9]. Our present aim is to prepare and compare the solvers, which were found the best from certain point of view, so now we consider the following ones.

Version 1 (Denoted by V1)

In [2,9], we prepared a number of variants of the original algorithm and the essence of the version which was found to be the most effective among the considered ones is the following.

In (2), we applied our spiral ordering for which the starting point was chosen as a node of nearly minimum eccentricity [10], from which a rooted level structure is generated. It is numbered by using the Cuthill-McKee's numbering which is reversed at the last.

In (3), we applied the IC(0) for preparing the preconditioner.

The PCG-solver in which the preconditioner is prepared by V1 was found more efficient than any of those considered in [1]. We refer to this solver as VIC(0).

Version 2 (Denoted by V2)

In [3,11], we considered the following versions of the original algorithm.

In (2), we applied our spiral ordering [9] for which the starting point was chosen as a node of minimum eccentricity, from which a rooted level structure is generated. It is numbered by using the Cuthill-McKee's numbering which is reversed at the last.

In (3), in preparing the preconditioner, we applied the level of fill-in dropping strategy [12,17], when the number of subdiagonals formed from the fill-ins is IP.

Clearly, in case of $IP = 0$, this V2 turns to V1. We tested V2 with the parameters $0 \leq IP \leq 10$, when some problems from the FEM structural analysis were solved. As it is discussed in [11], the PCG-solver in which the preconditioner is prepared by V2 with $IP = 2$ was found more efficient than any of those considered in [9]. We refer to this PCG-solver as VIC(2).

Now we continue preparing the following four versions as follows. In each version, we apply our spiral ordering, so for each one, it is enough to indicate the type of incomplete factorization applied in the algorithm.

Version 3 (Denoted by V3)

In (3), the preconditioning is prepared as MIC(0); that is, no fill-in is accepted, and all the discarded fill-in elements are added to the relevant diagonal entry. We refer to the relevant PCG-solver as VMIC(0).

Version 4 (Denoted by V4)

In (3), the preconditioning is prepared as MIC(2); that is, the accepted fill-in elements form two "subdiagonals", and all the discarded fill-in elements are added to the relevant diagonal entry. We refer to the relevant PCG-solver as VMIC(2).

Version 5 (Denoted by V5)

In (3), the preconditioning is prepared as RIC(0); that is, no fill-in is accepted, and all the discarded fill-in elements—modified by a relaxation parameter ω—are added to the relevant diagonal entry. We refer to the relevant PCG-solver as VRIC(0).

Version 6 (Denoted by V6)

The preconditioning is prepared as RIC(2); that is, the accepted fill-in elements form two subdiagonals and all the discarded fill-in elements—modified by a relaxation parameter ω—are added to the relevant diagonal entry. We refer to the relevant PCG-solver as VRIC(2).

Note that for V5 and V6—following [1]—for the relaxation parameter ω $(-1 \leq \omega < 1)$, we applied $\omega = 1 - h_0$, where $h_0 \approx 1/\sqrt[d]{\text{nuno}}$, where nuno denotes the number of nodes; d is the spatial dimension.

6. NUMERICAL EXPERIENCES AND CONCLUSIONS

We consider six PCG-solvers VIC(0), VIC(2), VMIC(0), VMIC(2), VRIC(0), and VRIC(2). To compare them by efficiency, we apply each one for solving the same problem, and we indicate the number of iterations required for achieving the same relative iterative error.

We consider six problems from the FEM structural analysis, when elastic frame structures with 3D beam elements are solved.

For an individual beam element, we consider the following differential equations in the local coordinate system (ξ, η, ζ):

$$
\begin{aligned}
I_\zeta E v^{(IV)}(\xi) &= p_\eta, \\
I_\eta E w^{(IV)}(\xi) &= p_\zeta, \\
A E u^{(II)}(\xi) &= p_\xi, \\
I_t G \varphi^{(II)}(\xi) &= \mu_\xi.
\end{aligned}
$$

The beam is supposed to lie on the axis ξ. Here (u, v, w) denote the relevant local displacements, respectively. $I_\eta = I_\zeta$, and they are the moments of inertia of the cross section. E is the Young's modulus. A is the cross section area, and p_ξ, p_η, and p_ζ are the distributed loads in the relevant three directions; μ_ξ is the distributed moment; I_t is the torsional moment of the cross-section; and G is the shearing modulus. From the equilibrium equation, the well-known system of linear equations is prepared whose coefficient matrix is the elementary stiffness matrix. We consider the square domain 1000 mm × 1000 mm in the (x, y) plane on which we generate a regular mesh consisting of cross-bars. It serves as the bar structure on which the system of linear equations is generated whose coefficient matrix is the whole stiffness matrix. For all the considered problems, the beam cross section is a circle with diameter d. Here, we have the parameters: $d = 20\,\text{mm}$, $A = 31416.10^2\,\text{mm}^2$, $I_\eta = I_\zeta = 7854.10^3\,\text{mm}^4$, $E = 2.10^6 N/\text{mm}^2$, $\nu = 0,3$, $G = E/2(1+\nu)$, where ν is the Poisson ratio.

We consider six problems and they differ from each other in the mesh parameter, that is, in the size of the problem. At the four corner point, the structure is fixed, and at the central point of the structure, a loading vector of $1000N$ is pointing outward from the bar structure in its normal direction.

The problems to be considered here differ from each others in the size. The spectral condition numbers of the tested stiffness matrices are large (10^6–10^7). For the relative iterative error, we used $\varepsilon = 10^{-4}$.

Notes on our implementation are as follows.

(a) The D-reduction was executed on the whole stiffness matrix (not on elementary stiffness matrices [1]).

(b) For each of the PCG-solvers, the ordering was executed on the connected components of the graph derived from the relevant matrix. The incomplete factorization was executed blockwise, by which a great reduction in computer time was achieved.

(c) For reducing the number of updatings during the incomplete factorization, we applied the concept of the IKJ-version of the Gaussian elimination due to Saad [17].

(d) We applied the overlay technique within the FORTRAN for trying to optimize the storage requirements.

Our computer results are presented in Table 1, where we indicate the number of iterations for all the considered solvers VIC(0), VIC(2), VMIC(0), VMIC(2), VRIC(0), and VRIC(2).

For the first problem being of the smallest size, all the considered problems we obtained no significant differences in the number of iterations. In [11], we have seen a similar tendency.

In comparing, VIC(0), VMIC(0), and VRIC(0), whose common property is that no fill-in was accepted ($IP = 0$) during the incomplete factorization process, VIC(0) is the most effective. For

Table 1. Computer results.

$n0$	n	VIC(0)	VIC(2)	VMIC(0)	VMIC(2)	VRIC(0)	VRIC(2)
1	702	45	54	56	52	50	51
2	2022	106	96	144	110	126	101
3	3342	188	158	247	188	217	175
4	4462	286	226	360	275	315	253
5	5982	393	307	482	369	420	337
6	7302	502	392	613	475	533	429

$n0$: serial number; n: size of the stiffness matrix.

VMIC(0), the number of iterations is larger than in the case of VRIC(0). That is, applying either MIC(0) or RIC(0), the effectiveness is worse than for IC(0), while in [1] both MIC(0) and RIC(0) were more efficient than IC(0) was (applied on regular grid). Similar to the case of [1], here RIC(0) is also somewhat better than MIC(0).

In comparing VIC(2), VMIC(2), and VRIC(2), whose common property is that two subdiagonals are produced from the accepted fill-in elements ($IP = 2$) during the incomplete factorization process, VIC(2) is found to be the most effective.

Table 2. Number of iterations for VIC(2).

$n \setminus b$	-4	-5	-6	-7	-8	-9	-10
707	54	63	75	82	89	96	102
2022	96	105	120	132	143	149	157
3342	158	177	202	213	229	240	252
4462	226	257	287	310	325	346	363
5982	307	357	385	412	436	457	471
7302	392	448	484	518	545	563	588

b: bounds for the relative iterative error;
n: size of the stiffness matrix.

The tendency of the numbers of iterations obtained is similar to the previous case, for all considered problems (except the first one). VRIC(2) is more efficient than VMIC(2). Applying VRIC(2), the number of iterations is about 5–11% larger than in the case of using VIC(2), while VRIC(2) is more efficient than VIC(0).

As a result, we can conclude that for a given relative iterative error, VIC(2) and RIC(2) were found to be the two most effective solvers among the considered ones.

Now, we consider the behavior of VIC(2) and VRIC(2) when decreasing the bound of the relative iterative error. In Tables 2 and 3, we present our results when $\varepsilon = 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$, $10^{-8}, 10^{-9}$, and 10^{-10} were used for VIC(2) and VRIC(2), respectively. In the tables, for each ε, we indicate only the corresponding exponent of the error bound which we denote here as b. Here we indicate also the number of iterations required for achieving the prescribed relative iterative error. From the tables, it can be seen that for both methods, by increasing the accuracy, the number of iterations is increased nearly linearly for all the considered problems.

As a result of the above discussion, we can conclude that version VIC(2) was found to be the most effective solver among the considered ones on our test problems.

Table 3. Number of iterations for VRIC(2).

$n \setminus b$	-4	-5	-6	-7	-8	-9	-10
707	51	62	76	82	90	94	101
2022	101	124	137	155	168	180	190
3342	175	204	230	258	281	302	320
4462	253	297	340	379	407	436	461
5982	337	401	452	503	540	586	611
7302	429	514	575	640	683	738	775

b: bounds for the relative iterative error;
n: size of the stiffness matrix.

REFERENCES

1. P. Saint-Georges, G. Warzee, R. Beauwens and Y. Notay, High-performance PCG solvers for FEM structural analysis, *Int. J. for Numer. Meth. in Engng.* **39**, 1313–1340, (1996).
2. I. Arany, Solving systems of elastic bar structures by preconditioned conjugate gradient method, *Computers Math. Applic.* **38** (9/10), 125–134, (1999).
3. I. Arany, Solving structural systems by some variants of the PCG-solver of Saint-Georges *et al.*, In *Proc. of 15. IMACS World Congress, Vol. 3*, (Edited by A. Sydow), pp. 443–448, Wissenschaft und Technik Verlag, Berlin, (1997).
4. J. Meijerink and H. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, *Math. Comp.* **31**, 148–162, (1977).
5. O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems, Theory and Computations*, Academic Press, New York, (1984).
6. O. Axelsson and N. Munksgaard, Analysis of incomplete factorizations with fixed storage allocations, In *Preconditioning Methods, Theory and Applications*, (Edited by D.J. Evans), pp. 219–241, Gordon and Breach, New York, (1983).
7. D.S. Kershaw, The incomplete Cholesky conjugate gradient method for the iterative solution of systems of linear equations, *J. of Comp. Phys.* **26**, 43–65, (1978).
8. I.S. Duff and G.A. Meurant, The effect of ordering on preconditioned conjugate gradients, *BIT* **29**, 635–657, (1989).
9. I. Arany, Using the PCG-solver by Saint-Georges *et al.* with spiral ordering in structural analysis, *Publ. Univ. Miskolc. Ser. D. Natural Sciences* **36**, 3–12, (1996).
10. I. Arany, An algorithm for finding a nearly minimum eccentricity in a graph, In *Proc. of Intern. Comp. Sci. Conf. Section K*, pp. 69–74, University Press, Miskolc, (1997).
11. I. Arany, On some variants of the PCG-solver due to Saint-Georges *et al.*, *Publ. Univ. Miskolc. Ser. D. Natural Sciences* **37**, 3–12, (1997).
12. J.W. Watts III, A conjugate gradient truncated direct method for the iterative solution of the reservoir simulation pressure equation, *Society of Petroleum Engineer Journal* **21**, 345–353, (1981).
13. I. Gustafsson, A class of first order factorization methods, *BIT* **18**, 142–156, (1978).
14. R. Beauwens, Modified incomplete factorization strategies, In *Lecture Notes in Math., Number 1457*, pp. 1–16, Springer-Verlag, New York, (1990).
15. O. Axelsson and G. Lindskog, On the eigenvalue distribution of a class of preconditioning methods, *Numer. Math.* **48**, 479–498, (1986).
16. Y. Notay, DRIC: A dynamic version of RIC method, *Numer. Lin. Alg. with. Appl.* **1** (6), 511–532, (1994).
17. Y. Saad, ILUT: A dual threshold incomplete LU factorization, Technical Report 92/38, University of Minnesota, Supercomputer Institute, (1992).
18. J.J. Júdice and J.M. Patrició, Truncated envelope preconditioning technique, *Commun. in Numer. Meth. in Engng.* **10**, 149–154, (1994).
19. Y. Notay, Ordering methods for approximate factorization preconditioning, Technical Report IT/IF/14–11, Université Libre de Bruxelles, (1993).
20. O. Axelsson and I. Gustafsson, Iterative methods for the solution of the Navier equations of elasticity, *Comp. Meth. in Appl. Mech. Enging.* **15**, 241–258, (1978).
21. S. Schlafman and I. Efrat, Using Korn's inequality for an efficient iterative solution of structural analysis problem, In *Iterative Methods in Linear Algebra*, (Edited by R. Beauwens and P. Groen), pp. 575–581, Elsevier Science, North Holland, Amsterdam, (1992).
22. O. Axelsson, On iterative solvers in structural mechanics, In *Proc. of 15. IMACS World Congress, Vol. 3*, (Edited by A. Sydow), pp. 431–436, Wissenschaft und Technik Verlag, Berlin, (1997).

23. O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, (1994).
24. R. Beauwens and R. Wilmet, Conditioning analysis of positive definite matrices by approximate factorizations, *J. Comp. and Appl. Math.* **26**, 257–269, (1989).
25. P. Saint-Georges, *Iterative Solution of Linear Systems for FEM Structural Analysis*, University of Libre de Bruxelles, Faculty of Applied Sciences, (1996).